PSis - Programação de Sistemas - 2021/2022
1st Exam, 10th February 2022, 18h00, Duration: 1h30

Systems [1.0]

1. [1.0 points] When defining a complex system using a systemic approach, one principle is that a system can be broken into smaller subsystems and these sub-systems interact between each other using interfaces.

a) Present two distinct types/classes of interfaces that sub-systems can implement, and that the programmer can use to link those sub-systems.
b) For each type of interface, present the scope of usage (local/remote) and the major concerns that should be considered when implementing and using such interfaces.

API / Libraries
- local
- verification of the arguments

Communication channels/messages –
- remote
- verification of the messages and communication errors

Shared memory
- Local
- synchronization

Concurrency [2.0]

2. [1.0 points]

a) Describe what is Concurrency.
b) Describe how concurrency can be implemented in an Operating System with only one CPU

capacity of the execution of two tasks that execute during the same period and access/use a single resource

With only one CU the Operating System, assigns each task a small time-slice.
During this time-slice the task is executing on the CPU (the others are waiting) after the end of the time-slice new task starts executing in the CPU.

3. [1.0 points]

a) Present and describe two types of parallelism.

b) For each one, present one example of an implementation of such type.

Data-parallelism –

execution of the same code/task in different data at the same time

execute the same transformation into multiple files

task parallelism –

execution of different task as the same time

running the excel at the same time that a video is being processed

Threads / Synchronization [3.0]

4. [1.0 point] Suppose that a program reads lines from a file and for each line creates a new thread immediately after reading it.

a) Write the C/pseudo code that reads text lines from a previously opened file and creates a new thread for each line immediately after reading it. Each thread should receive as argument a different line from the file.

b) Write the code inside the thread that prints the received line length.

Guarantee that the relevant arguments for pthread_create and the thread code are correct.

```
 while(xxx){
    fgets(fp, line)
    new_line = malloc(strlen(line))
    strcpy (new_line, line)
    pthread_create (processing_functio, & new_line)
 }

void * processing_functin(void * arg){
   printf(strlen(arg))
   ...
```

5. [1.0 point]

- Explain what happens to the program (and other threads) when a thread erroneously terminates before unlocking a lock (inside a critical region).

With regular mutex, the mutex that was locked by the thread remains locked.
When other threads try to enter the critical region they will remain waiting to enter the critical region.
Since the thread that locked the mutrex is dead the mutex will never be unlocked.
The program will eventually enter into a deadlock

6. [1.0 point] Multiple threads can access global arrays without requiring synchronization if each thread accesses different array positions.

a) Describe one problem that may happen when different threads access distinct positions on the same array.
b) Explain in detail the type/pattern of accesses and the consequences to the program execution.

False sharing
when a thread writes into a position of the array the whole cache line is invalidated
other threads accessing near position will require the reading from memory an not cache

if these write are frequent, the number of invalidates and misses will be high and the program will run slower

System calls / Processes [2.0]

7. [1.0 point] System calls are different from the regular C libraries function.

a) Describe how a system call is executed from the moment it is called.
b) Give one example of one system call and one C library function that have a similar objective. Explain how they are related and different.

A small part of the system call runs as regular C code of a program but then the control is given to the Kernel/Operating system.
The kernel will execute the system call in a privileged mode and at the end return the result back to the applications
system-call – read
C library function - fread

8. [1.0 point] Processes were the first mechanism that implemented concurrency and parallelism in modern operating systems, but are still used today when implementing servers.

- Describe why most mail servers running on Linux create a new process after receiving the connection of a client, instead of creating a new thread.

Since processes are isolated from each other, bug in the code of a process does not affect others: a segmentation fault only terminates a process while a segmentation fault in a thread terminates all other threads and it is impossible to access the data of another user on another processed

Inter-process communication (IPC) [2.0]

9. [1.0 point] Shared memory is one of the types of IPC mechanisms, sockets (datagram and stream) are another.

a) Present one other IPC mechanism that allows communication between processes in the same machine.
b) Describe one difference between the presented IPC mechanism and shared memory.
c) Describe one difference between the presented IPC mechanism and sockets.

Pipes pipes do not require synchronization while shared memory requires pipes only offer half-duplex communication, while sockets offer full duplex

10. [1.0 point] In the project, students changed datagram sockets into stream sockets when implementing the new-super-pong.

a) Describe two advantages of using stream sockets over the use of datagram sockets.
b) Present one major change in the organization of a program when changing the communication from data-gram sockets to stream sockets.

Each stream socket is associated with only tow entities it is possible to send large data , offer reliability and ordering guarantees requires the creation of a thread per client